ECE540: Homework #4

David Kirby

Due: 08 November 2020

Chapter 8 (7^{th} Edition)

Review Questions

Section 8.1

1. R1. What are the differences between message confidentiality and message integrity? Can you have confidentiality without integrity? Can you have integrity without confidentiality? Justify your answer.

   Confidentiality means that if the intruder intercepts a ciphertext message, they cannot determine the plaintext message. Message integrity means that the receiver can detect if the message was tampered with (plaintext or ciphertext). Yes, it's possible to have confidentiality without integrity (the message could be altered by other means, but the intruder still would not be able to determine the plaintext). Yes, it's possible to have integrity without confidentiality (sending a message as plaintext is an example of this).

Section 8.2

2. R3. From a service perspective, what is an important difference between a symmetric-key system and a public-key system?

   In a symmetric-key system, both parties share a secret key. In a public-key system there are public and private keys, meaning different encryption and decryption keys.

3. R4. Suppose that an intruder has an encrypted message as well as the decrypted version of that message. Can the intruder mount a ciphertext-only attack, a known-plaintext attack, or a chosen-plaintext attack?

   A known-plaintext attack since the intruder has both the encrypted and decrypted version of the message.

4. R5. Consider an 8-block cipher. How many possible input blocks does this cipher have? How many possible mappings are there? If we view each mapping as a key, then how many possible keys does this cipher have?

   2^8 possible input blocks; $2^8!$ possible mappings; and $2^8!$ possible keys.

5. R7. Suppose $n = 10,000,\ a = 10,023,\ b = 10,004$. Use an identity of modular arithmetic to calculate in your head $(a \cdot b) \bmod n$.

$$(a \cdot b) \bmod n = (a \bmod n) \cdot (b \bmod n) = 23 \cdot 4 = 92$$

Problems

1. P1. Using the monoalphabetic cipher in Figure 8.3, encode the message "This is an easy problem." Decode the message "rmij'u uamu xyj."

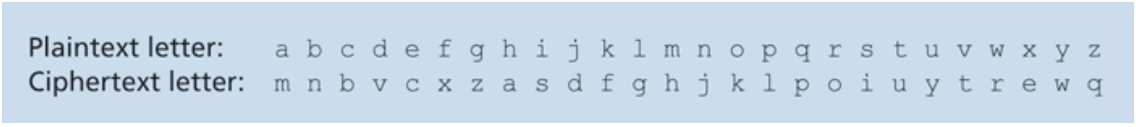


Figure 8.3: A monoalphabetic cipher

Encoded message: `Uasi si mj cmiw lokngch.`
Decoded message: `wasn't that fun.`

2. P2. Show that Trudy's known-plaintext attack, in which she knows the (ciphertext, plaintext) translation pairs for seven letters, reduces the number of possible substitutions to be checked in the example in Section 8.2.1 by approximately 10^9.

   Normally, the monoalphabetic ciphertext pairs would be 26!, but with 7 letters known, this brings the pairs to 19!. The difference between them would be $26 \cdot 25 \cdot 24 \cdot 23 \cdot 22 \cdot 21 \cdot 20 \approx 10^9$.

3. P9. In this problem, we explore the Diffie-Hellman (DH) public-key encryption algorithm, which allows two entities to agree on a shared key. The DH algorithm makes use of a large prime number p and another large number g less than p. Both p and g are made public (so that an attacker would know them). In DH, Alice and Bob each independently choose secret keys, S_A and S_B, respectively. Alice then computes her public key, T_A, by raising g to S_A and then taking mod p. Bob similarly computes his own public key S_B by raising g to S_B and then taking mod p. Alice and Bob then exchange their public keys over the Internet. Alice then calculates the shared secret key S by raising T_B to S_A and then taking mod p. Similarly, Bob calculates the shared key S' by raising T_A to S_B and then taking mod p.

   (a) Prove that, in general, Alice and Bob obtain the same symmetric key, that is, prove $S = S'$.

$$\begin{aligned} S &= T_B^{S_A} \bmod p \\ &= \left(g^{S_B} \bmod p\right)^{S_A} \bmod p \\ &= \left(g^{S_A} \bmod p\right)^{S_B} \bmod p \\ &= T_A^{S_B} \bmod p \\ &= S' \end{aligned}$$

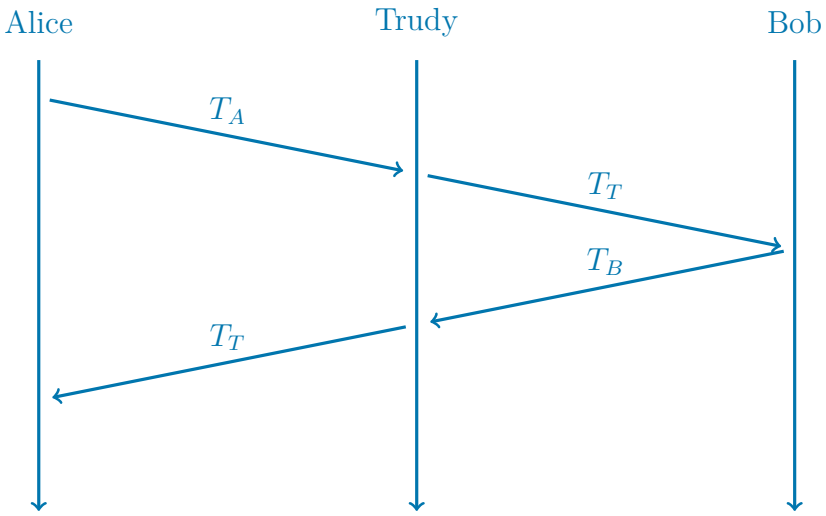
   (b) With $p = 11$ and $g = 2$, suppose Alice and Bob choose private keys $S_A = 5$ and $S_B = 12$, respectively. Calculate Alice's and Bob's public keys, T_A and T_B. Show all work.

$$\begin{aligned} T_A &= g^{S_A} \bmod p \\ &= 2^5 \bmod 11 \\ &= 10 \end{aligned} \qquad \begin{aligned} T_B &= g^{S_B} \bmod p \\ &= 2^{12} \bmod 11 \\ &= 4 \end{aligned}$$

(c) Following up on part (b), now calculate S as the shared symmetric key. Show all work.

$$\begin{aligned} S &= T_B^{S_A} \bmod p \\ &= 4^5 \bmod 11 \\ &= 1 \end{aligned}$$

(d) Provide a timing diagram that shows how Diffie-Hellman can be attacked by a man-in-the-middle. The timing diagram should have three vertical lines, one for Alice, one for Bob, and one for the attacker Trudy.



Trudy intercepts Alice's public key and sends her own key to Bob instead. When Bob sends his key to Alice, Trudy intercepts that as well, meaning the Trudy can now has both keys and can generate the shared key between Bob and Alice.

4. P12. Suppose Alice and Bob share two secret keys: an authentication key S_1 and a symmetric encryption key S_2. Augment Figure 8.9 so that both integrity and confidentiality are provided.

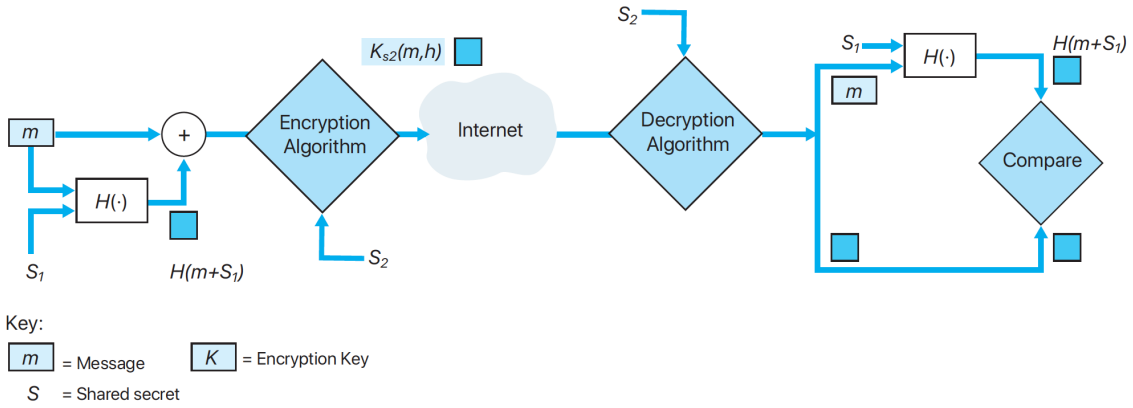


Figure 8.9: MAC augmented with symmetric encryption key S_2

Please see revised Figure 8.9 above for addition of symmetric encryption key S_2.

5. P15. Consider our authentication protocol in Figure 8.18 in which Alice authenticates herself to Bob, which we saw works well (i.e., we found no flaws in it). Now suppose that while Alice is authenticating herself to Bob, Bob must authenticate himself to Alice. Give a scenario by which Trudy, pretending to be Alice, can now authenticate herself to Bob as Alice. (*Hint*: Consider that the sequence of operations of the protocol, one with Trudy initiating and one with Bob initiating, can be arbitrarily interleaved. Pay particular attention to the fact that both Bob and Alice will use a nonce, and that if care is not taken, the same nonce can be used maliciously.)

   This is similar to modern-day catfishing where an intruder pretends to be someone else. In this case, Trudy initiates contact with Bob by pretending to be Alice; Bob authenticates himself by sending his and Alice's shared key; Trudy cannot reply yet as she does not know the key. Bob sends a nonce, Trudy uses that as her response back to Bob, and Bob essentially authenticates himself using his own key instead of Alice's. This now gives Trudy the nonce shared key. She can now reply back to Bob with this nonce key making Bob think she is Alice.

6. P19. Consider the Wireshark output below for a portion of an SSL session.

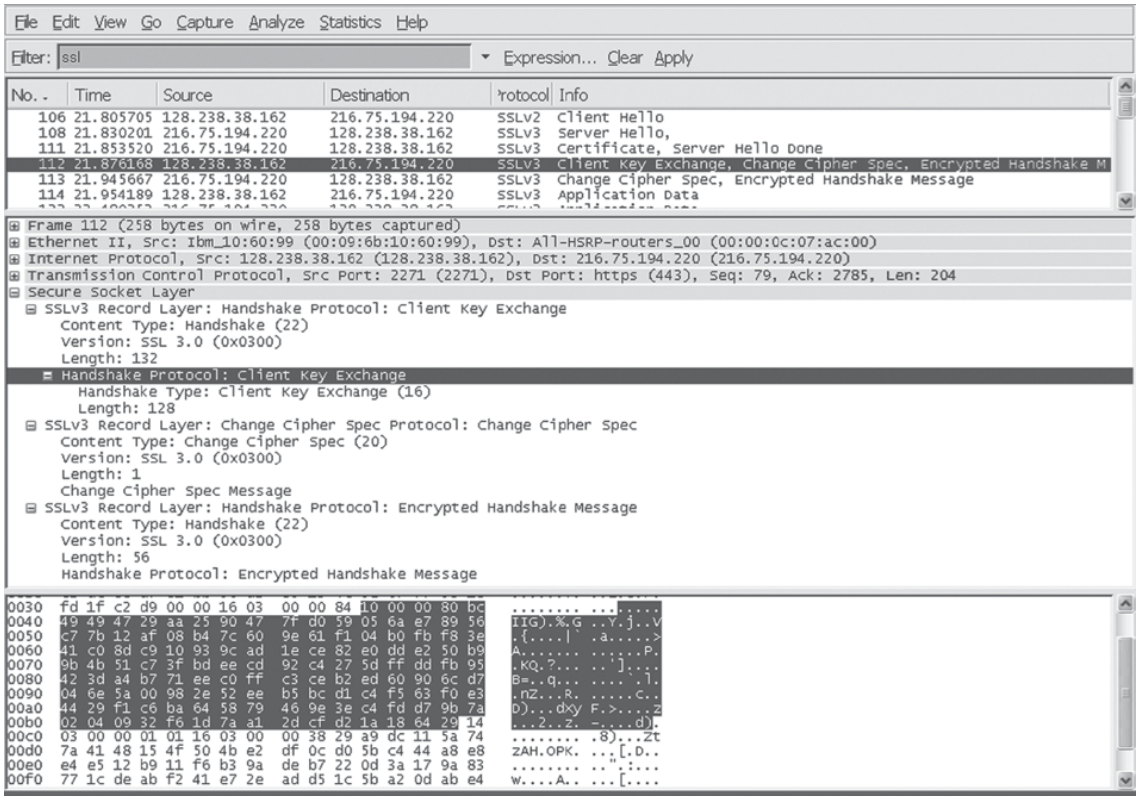


Figure 8.0: (Wireshark screenshot reprinted by permission of the Wireshark Foundation.)

(a) Is Wireshark packet 112 sent by the client or server?

Client

(b) What is the server's IP address and port number?

IP: `216.75.194.220`
Port: `443`

(c) Assuming no loss and no retransmissions, what will be the sequence number of the next TCP segment sent by the client?

Sequence #: 283

(d) How many SSL records does Wireshark packet 112 contain?

Packet 112 contains 3 SSL records.

(e) Does packet 112 contain a Master Secret or an Encrypted Master Secret or neither?

Packet 112 contains an Encrypted Master Secret.

(f) Assuming that the handshake type field is 1 byte and each length field is 3 bytes, what are the values of the first and last bytes of the Master Secret (or Encrypted Master Secret)?

First byte: `bc`
Last byte: `29`

(g) The client encrypted handshake message takes into account how many SSL records?

Client encrypted handshake: 6 SSL records

(h) The server encrypted handshake message takes into account how many SSL records?

Server encrypted handshake: 9 SSL records

7. P20. In Section 8.6.1, it is shown that without sequence numbers, Trudy (a woman-in-the middle) can wreak havoc in a TLS session by interchanging TCP segments. Can Trudy do something similar by deleting a TCP segment? What does she need to do to succeed at the deletion attack? What effect will it have?

Yes, Trudy could achieve the same chaos as without sequence numbers. In order for deleting segments to work, Trudy would need to adjust the sequence numbers accordingly, otherwise Bob and Alice would notice that they did not receive the correct number of segments. If Trudy could accomplish sequence renumbering, Bob and Alice would be receiving a stream with missing packets and never even know it.